Interspeech 2024
1-5 September 2024, Kos, Greece

Homograph Disambiguation with Text-to-Text Transfer Transformer

Markéta Řezáčková[1,2], *Daniel Tihelka*[1], *Jindřich Matoušek*[1,2]

[1]New Technologies for the Information Society (NTIS) and [2]Department of Cybernetics, Faculty of applied Sciences, University of West Bohemia, Pilsen, Czech Republic

{juzova,dtihelka,jmatouse}@ntis.zcu.cz

Abstract

In recent years, the Text-to-Text Transfer Transformer (T5) neural model has proved very powerful in many text-to-text tasks, including text normalization and grapheme-to-phoneme conversion. In the presented paper, we fine-tuned the T5 model for the task of homograph disambiguation, which is one of the essential components of text-to-speech (TTS) systems. To compare our results to those of other studies, we used an online dataset of US English homographs called Wikipedia Homograph Data. We present our results, which outperformed the previously published single-model approaches. We also focus on more detailed error analysis, model performance on different types of homographs, and the impact of training set size on homograph disambiguation.
Index Terms: homograph disambiguation, word-sense disambiguation, text-to-text transfer transformer, text normalization, speech synthesis

1. Introduction

One of the key components in text-to-speech (TTS) systems is the natural language processing model, which, among many other tasks, is responsible for homograph disambiguation. In general, homographs are words sharing the same written form but differing in pronunciation [1]. In the task of assigning the correct pronunciation, the lexical semantic context may be used to disambiguate lexical homographs like *bass* (fish vs. musical instrument, both nouns), or POS tags are utilized to disambiguate morphosyntactic (e.g., nominal vs. verbal) class of homographs. However, POS tagging is prone to errors, and an additional set of rules/dictionary is required for the whole system to work correctly; there is an effort to tackle this problem using modern DNN approaches, e.g., [2, 3]. This also applies to this paper, where the T5-based model is challenged on the well-defined *Wikipedia Homograph Data* dataset [4]. In any case, the wrong variant determination of a particular homograph can lead to misunderstanding of the synthesized text, or at least the intelligibility and naturalness of the generated speech is corrupted.

Regarding the terminology, there are many homograph categories, with noun / verb being the most frequent. One or more different homograph words may belong to a category, e.g., *record, tear, upset*. Each word then create corresponding variants `rEk3:d`[1](noun) / `r@kO:rd`(verb) with different pronunciations. Each of the variants has then several samples in the dataset (e.g., *record*(noun) has 78 samples in the training set and 10 samples in the evaluation set), as shown in Table 1.

As indicated earlier, we also distinguish two different homograph types, respecting the broad source of homography – morphosyntactic derivation of the same lemma (also called *part-of-speech homographs*), or lexically distinct terms (also called *accidental homographs*; the same written form is a consequence of pure coincidence, usually caused by different word origins from other languages) [1]. The former group of homographs contains e.g. the words *record* and *read*, while *defect*, *bass*, *minute*, and *lead* represent lexical homographs. The homographs *subject* and *live* are examples of combining the previous terms.

All homographs mentioned in the previous paragraphs are examples of so-called *true homographs*, whose ambiguity is not caused by writing style or abbreviation but rather due to their spelling [1]. Furthermore, there are, for example *abbreviation homographs* (e.g., *Dr* meaning drive/doctor) – but those are not taken into account in this study.

Determining a specific variant of a given homograph is also called a *word-sense disambiguation* (WSD), which is characterized by assigning labels to only some words in the sentence (contrary to *tagging* when a label is given to all words in the input sequence). The very first attempts at solving homograph ambiguity were rule-based approaches (with both manually and automatically derived rules) taking into account only the immediate (word) context of the examined word or searching a *trigger* in the sentence [1, 6]. Later, different types of classifiers, decision trees, etc., were used, utilizing word and POS features (e.g. [7]). Nowadays, more complex solutions involving different types of deep learning methods have been presented – see Section 3.

2. Data and Model Description

To be able to compare our approach with the others meaningfully, we work with *Wikipedia Homograph Data* [4, 8] (WHD), an online available dataset containing 162 unique homograph words (*read, live, ...*) with roughly 100 samples per homograph variant (in US English). In total, it contains almost 16 thousand sentences, which are split into a training set (90%) and an evaluation set (10%), each sentence having exactly one homograph word. There are 2 homograph variants for almost all homographs words, except for two of them having three variants (homographs *august* and *mobile*, both lexical homographs). The examples for the homograph *august* are listed below:

- noun (month) `[A:g@st]` – *Peak birding season is **August** to October.* (76x in training set)
- proper noun `[aU:gUst]` – *It was founded in 1885 by master builder **August** Mayer.* (8x in training set)
- adjective `[A:gVst]` – *May Nusku, the **august** vizier, hear my prayer and intercede for me.* (1x in training set)

All homographs are also coded by a homograph type (either morphosyntactic, lexical, or mixed) – which would be used for detailed evaluation in Section 4.2.

The basic statistics of the training set are shown in Table 2.

[1]All examples of pronunciation are in the SAMPA alphabet [5].

10.21437/Interspeech.2024-949

Table 1: *Homograph variant examples. The brackets show the number of occurrences in the training/evaluation WHD set.*

Category	Homograph word	Variant 1		Variant 2	
noun / verb	record	[rEk3:d]	(78x / 10x)	[r@kO:rd]	(12x / 0x)
	defect	[di:fEkt]	(79x / 10x)	[d@fEkt]	(11x / 0x)
adjective-noun / verb	subject	[sVbdZEkt]	(90x / 10x)	[s@bdZEkt]	(0x / 0x)
adjective / verb	live	[laIv]	(54x / 6x)	[lIv]	(33x / 4x)
noun / noun	bass	[bes] (music)	(72x / 10x)	[b{s] (animal)	(13x / 0x)
adjective / noun	minute	[maInu:t]	(12x / 1x)	[mIn@t]	(74x / 9x)
noun / noun-verb	lead	[lEd] (material)	(15x / 3x)	[li:d]	(82x / 8x)
verb / verb	read	[ri:d] (present tense)	(59x / 6x)	[rEd] (past tense)	(52x / 7x)

The individual homograph words are distributed relatively uniformly, with an average of 88.03 samples for each word (standard deviation equal to 3.98) in the training set (roughly 10 in the evaluation set, therefore). However, the homograph variants for a word are not distributed so uniformly, on average with 85.1% samples for one of the variants. The minimum is 58 samples for *dove*, from which 52 are for the noun (bird) and 6 for the past tense verb. Conversely, *read* has 111 samples, from which 59 are for present and 52 for past tenses. The extreme case is the word *object* with 89 noun variants but no verb variant in both the training and evaluation sets.

Table 2: *Train data statistics. The second column shows the distribution of homograph words through the dataset; the third column is the distribution of homograph variants in samples for a word.*

Statistical value	No. of samples per homograph words	Percentage of more frequent variant
mean	88.03	85.10
standard deviation	3.98	14.68
maximum	111	100.00
minimum	58	50.00

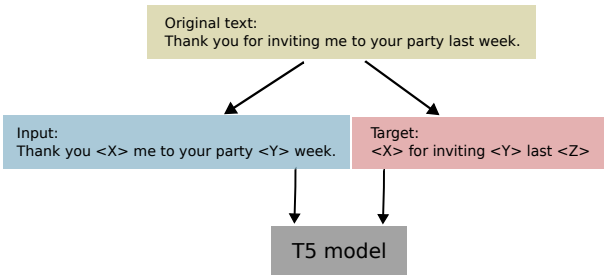

Figure 1: *Example of processing the original text and creating input/target text pairs for T5 training (inspired by [9]).*

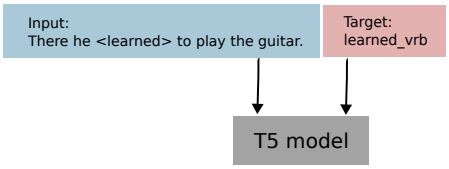

Figure 2: *Example of creating input/target text pairs for T5 fine-tuning process to the homograph disambiguation task.*

2.1. Our T5-based model for homograph disambiguation

As an alternative to the previous homograph disambiguation approaches mentioned, we used the Text-to-Text Transfer Transformer (T5) model, a self-supervised trained variant of the generic textual Transformer encoder-decoder architecture [9]. The reason is that in the case of the T5 model, the same objective, training procedure, and decoding process can be utilized for a wide variety of NLP problems, such as phonetic transcription [10], question answering [11], document summarization, and others [9].

The generic T5 model is pre-trained on an artificial self-supervised task – a text restoration task from unlabelled training data. During this, the model tries to recover missing tokens in the input sentence masked with sentinel tokens <X> and <Y>. The masked tokens are used as training targets, and the output sentence is terminated using another sentinel token <Z>, as illustrated in Figure 1. This way, T5 learns not only the knowledge required to understand the input sentence but also the knowledge necessary to generate meaningful output sentences.

For our experiment with English homograph data, we used the Google's T5-base English model (version from October 2019, [12]) as a generic model. It was trained from Common Crawl data [13]. The pre-processing steps for building the Colossal Clean Crawled Corpus (C4) are described in [9]. We used the same settings as our previous experiments with the T5 model regarding the text normalization and G2P tasks (e.g., [10, 14]). The `t5-base` architecture thus consists of 220M parameters, 2×12 transformer block in the encoder and the decoder. The dimensionality of hidden layers and embeddings was 768. The attention mechanism uses 12 attention heads with inner dimensionality of 64.

The second step was fine-tuning Google's T5 model to the target task of homograph disambiguation. As the T5 model is an encoder-decoder model that converts all NLP problems into a text-to-text format, the whole sentence from the WHD dataset, with the homograph highlighted, was the input during the training (fine-tuning), and a homograph ID was the expected output, as illustrated in Figure 2. TensorFlow implementation of Hugging-Face Transformers library [15], together with the T5s library [16], was used, which simplifies mainly the training and prediction processes by accepting simple tab-separated input-target pairs. The T5s library also uses variable sequence length and variable batch size to fully utilize the underlying GPU used for training.

The homograph disambiguation fine-tuning process lasted 100 epochs, with 1000 update steps per epoch. We used the Inverse Square Root Scheduler as the learning rate scheduler with a starting learning rate equal to 10^{-4}. All parameters were set as trainable.

3. Related work

In the present paper, we compare the use of T5 to different approaches using the same dataset. Their primary specifics are described in the following paragraphs.

The authors of the WHD dataset [4] compare a rule-based approach (based on word context and POS tag rules) to a machine learning (ML) approach based on a set of multinomial log-linear classifiers. Those classifiers (one per homograph) are fed with several features – POS tag of the current homograph, capitalization feature (i.e., uppercase / title case / lowercase), and word context (preceding and following words). Subsequently, they are trained to distinguish between the homographs variants. The authors also tested the combination of both approaches and yielded very high accuracy values on evaluation data (see Table 3).

Another study [17] compares several part-of-speech taggers and inspects the improvements in noun-verb ambiguity when using word embeddings and data augmentation. They also use the WHD dataset to show the impact of their best model on a downstream task, and they outperformed the original approach [4] by using a tagger presented in [18] enhanced with contextual word embeddings.

The paper [19] also describes the use of per-homograph classifiers – they trained logistical regression models and fed them by transformer-based contextual word embeddings (comparing the utilization of more BERT-like [20] pre-trained models). The best results outperformed the previous approaches, and the subsequent data augmentation process (manual creation of new sentences for several words with lower accuracy and a highly unbalanced training set) even improved the results a little.

The main specific of *SoundChoice* [2], a grapheme-to-phoneme (G2P) model based on an encoder-decoder architecture, is the embedded homograph loss that penalizes errors made on homograph words (using the WHD dataset). It operates at the sentence level, similarly to e.g. [10] (although the vast majority of G2P approaches are still based on a dictionary, word-level approach). Focusing for now only on the homograph disambiguation task, the published results are slightly worse (compared to previously described studies), but it must be emphasized that a single model for all homographs was trained (which, of course, makes the whole process easier).

Another sentence-level G2P model [3] was trained using a 12-layer transformer encoder-decoder model (inspired by [21]) in both monolingual and multilingual (based on 24 languages) modes and tested on the WHD dataset (see the Table 3).

4. Results and discussion

For the evaluation of the fine-tuned T5 model capability in homograph disambiguation, the remaining 10% of the English homograph dataset were used. Having the same evaluation set as all the other approaches, we can directly compare the results obtained. As a measure, the micro-accuracy (a percentage of all correctly classified samples) and macro-accuracy (an arithmetic mean of accuracies counted per homograph word) were used in most of the papers from Section 3. We use the former in our paper and call that just *accuracy*, as used in [2, 3]. We decided to omit the macro accuracy, as it showed more or less identical numbers as the accuracy (micro-accuracy) in our experiment, which is also consistent with the results presented in the other papers.

The results for the whole evaluation set are shown in Table 3. Our T5 model is by far the best compared to the approaches using a single model for all the homograph words. The results better than ours are only in the experiments employing specialized per-homograph classifiers, or *rules + classifiers* [4] where special hand-tuned rules were designed to achieve the accuracy 99%. This is a surprisingly positive result, considering the size of the dataset not being exceptionally large for the T5 model and that no data augmentation technique was used. We expect even better results with a more extensive (and possibly more balanced, see Section 2) dataset or with an augmentation employed.

Table 3: *Results of Wikipedia Homograph Data disambiguation.*

Model/Approach	**Accuracy [%]**
Models for each word separately:	
[4] rules	89.0
[4] classifiers	95.4
[4] rules + classifiers	99.0
[17] tagger + embeddings	96.7
[19] classifiers + embeddings	99.1
[19] + data augmentation	99.3
One model for all words:	
[2] SoundChoice	94.0
[3] monolingual	94.5
[3] multilingual	95.9
our T5-based	**97.9**

4.1. Error analysis

From the 162 homograph words, 139 achieved 100% accuracy, and 17 homograph words achieved an accuracy of at least 86%, corresponding to a single error in the evaluation data. Only 6 of the homographs listed in Table 4 have accuracy lower than 85% (2 or more errors).

The detailed look at the particular words did not reveal any error pattern. Therefore, we summarize our findings as follows:

bow 2 out of 4 in the noun(knot) variants and 1 out of 6 in the noun(ship) are wrong. They are almost balanced in the training set.

decrease 2 out of 5 verbs are wrong, while both verb and noun types are balanced in the training dataset, similar to the previous word.

content 3 errors, all in the adj-noun-verb category; this category has only 1 sample in the training data, though, so there is no surprise in the low accuracy.

tear 3 out of 9 verb variants are wrong, while this type is even more frequent in the training data than the noun variant.

upset 2 out of 3 verbs are wrong; here, this type is less frequent in the training data.

reading 2 samples are incorrectly assigned to the noun(city) variant, but most likely due to their first letter being capitalized as part of book names.

Unfortunately, the other authors do not describe their proposed classifiers' failures in detail, so we cannot compare whether the errors made by T5 belong to a similar category as the other classifiers or are unrelated throughout the approaches. The first option would suggest that these types of errors could be avoided by increasing the dataset size. At the same time, the second would probably mean that the T5 is unable to capture a property that the other classifiers are (and vice versa).

Table 4: *Words with lower accuracy. The brackets show the numbers of occurrences in the training/evaluation sets.*

Homograph word	Variant 1		Variant 2		Accuracy [%]
bow	noun (ship) [baU]	(47x / 6x)	noun (knot) [boU]	(42x / 4x)	60
decrease	noun [di:kri:s]	(41x / 5x)	verb [d@kri:s]	(48x / 5x)	80
content	noun (filling) [kA:ntEnt]	(85x / 7x)	adj-noun-verb [k@ntEnt]	(1x / 3x)	70
tear	noun [ti:r]	(32x / 1x)	verb [tEr]	(57x / 9x)	70
upset	noun ['VpsEt]	(64x / 5x)	verb [Vp'sEt]	(24x / 3x)	75
reading	noun / verb [ri:d@N]	(71x / 8x)	noun (city) [rEd@N]	(16x / 2x)	80

4.2. Results for different homographs types and categories

As described in Section 2, we distinguish three basic types of homographs – morphosyntactic and lexical, and some homographs belong to the combined type. The numbers of homograph words belonging to these types in the training set are listed in the 2nd column of Table 5. While morphosyntactic homographs always differ in their part-of-speech category, the POS category could be the same (or very close) for those in the lexical type – 19% of our lexical homograph words belong to the same morphological class, mostly nouns.

This is likely the case of the much lower accuracy for lexical homographs, compared to the accuracy of those of morphosyntactic type, as shown in Table 5, even when the number of homograph words does not differ significantly. Although the T5 model does not explicitly learn any POS tags during both pretraining and fine-tuning processes, when learning the knowledge of the language, some morphological representation is probably learned internally. And the T5 model could benefit from that in this experiment.

Table 5: *Homograph disambiguation per homograph type*

Honograph type	No. of homographs	Accuracy [%]
morphosyntactic	78	98.1
lexical	62	97.1
combination	22	99.5

A similar trend can also be seen in the Table 6, showing the results of homograph disambiguation for individual categories – the decision-making between two homograph variants with the same morphological class evinces more errors, which is evident from the lower values for e.g. noun-noun or verb-verb combinations. On the other hand, the noun-verb, adjective-verb, and adjective-noun categories have very high accuracy values.

Table 6: *Homograph disambiguation per different categories*

Homograph category	No. of homographs	Accuracy [%]
noun / verb	101	98.5
adjective-noun / verb	26	98.0
adjective / verb	12	99.2
noun / noun	7	93.9
adjective / noun	6	100.0
noun / noun-verb	5	93.6
verb / verb	1	92.3
other categories	4	90.0

4.3. The impact of training set size

Since the number of homograph samples is not uniformly distributed in the WHD dataset, ranging from 58 (*dove*) to 111 (*read*) samples, as described in Section 2, we looked at the performance of the proposed T5 model with lower and more balanced samples in individual variants. We started with 10 samples per homograph word (sentences in the training set), where the samples were selected randomly from the training set. The aim was to balance the individual variants of each word (5 samples in both variant 1 and variant 2), but when not enough samples were available for a particular variant, all were kept, and the set was completed from the other variant. For 10 samples, 58 homographs still had less than 5 samples in one of the variants. We increased the number of samples to as many as 58, which was the maximum number of samples available if we wanted to have a balanced number of samples per word. The evaluation dataset was left untouched.

While the accuracy increases with the number of samples, the increase is only 0.2% percentage point since 30 samples (accuracy 97.3% to 97.9%). Even with 50 samples, there are only 3 more errors on the evaluation set (accuracy 97.7% to 97.9% for the full train set), while the reduced train set has only 8,100 sentences. It can be seen that the T5 model is rather capable of dealing with the disambiguation even when fewer data are provided to it.

Table 7: *Accuracy for the training set size reduction*

No. of samples per homograph	No. of homographs with imbalanced variants	Accuracy [%]
10	58	89.9
20	86	96.7
30	111	97.3
40	118	97.5
50	129	97.7
58	158	97.9

5. Conclusion and future work

We have shown that the universal T5 model fine-tuned to the target task of homograph disambiguation can easily outperform the other approaches on the same dataset, without any need for special treatments or tweaks of the model. While some of the approaches obtained even higher accuracy, they used specialized per-homograph models or a hand-tuned set of rules.

In our future work, we would like to try to yet increase the accuracy by extending the dataset with homograph words with the lowest accuracy, by balancing most of the unbalanced samples, and/or by augmenting the training dataset. Also, the ability to directly determine the phonetic variant of the homograph (or of the whole input sentence), and the handling of more than a single homograph per sentence would be useful for a practical use.

6. Acknowledgements

This research was supported by the Czech Science Foundation (GA CR), project No. GA22-27800S, and by the grant of the University of West Bohemia, project No. SGS-2022-017. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

7. References

[1] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[2] A. Ploujnikov and M. Ravanelli, "SoundChoice: Grapheme-to-Phoneme Models with Semantic Disambiguation," in *Proc. Interspeech 2022*, Incheon, Korea, 2022, pp. 486–490.

[3] G. Comini, S. Ribeiro, F. Yang, H. Shim, and J. Lorenzo-Trueba, "Multilingual context-based pronunciation learning for Text-to-Speech," in *Proc. Interspeech 2023*, Dublin, Ireland, 2023, pp. 631–635.

[4] K. Gorman, G. Mazovetskiy, and V. Nikolaev, "Improving homograph disambiguation with supervised machine learning," in *Proc. International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018.

[5] J. C. Wells, "SAMPA computer readable phonetic alphabet," in *Handbook of Standards and Resources for Spoken Language Systems*, D. Gibbon, R. Moore, and R. Winski, Eds. Berlin and New York: Mouton de Gruyter, 1997.

[6] D. Yarowsky, "Homograph disambiguation in text-to-speech synthesis," in *Progress in Speech Synthesis*, J. P. H. van Santen, J. P. Olive, R. W. Sproat, and J. Hirschberg, Eds. New York, NY: Springer New York, 1997, pp. 157–172.

[7] E. Black, "An experiment in computational discrimination of english word senses," *IBM Journal of Research and Development*, vol. 32, no. 2, pp. 185–194, 1988.

[8] K. Gorman, V. Nikolaev, and G. Mazovetskiy, "Homograph disambiguation data," 9 2021. [Online]. Available: https://github.com/google-research-datasets/WikipediaHomographData

[9] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[10] M. Řezáčková, J. Švec, and D. Tihelka, "T5G2P: Using Text-to-Text Transfer Transformer for grapheme-to-phoneme conversion," in *Proc. Interspeech 2021*, Brno, Czech Republic, 2021, pp. 6–10.

[11] J. Švec, M. Bulín, A. Frémund, and F. Polák, "Asking questions: an innovative way to interact with oral history archives," in *Proc. Interspeech 2023*, Dublin, Ireland, 2023, pp. 3679–3680.

[12] "T5: Text-To-Text Transfer Transformer," 10 2019. [Online]. Available: https://github.com/google-research/text-to-text-transfer-transformer

[13] "Common Crawl," 2017. [Online]. Available: https://commoncrawl.org/

[14] M. Řezáčková, A. Frémund, J. Švec, and J. Matoušek, "T5G2P: Multilingual grapheme-to-phoneme conversion with text-to-text transfer transformer," in *Pattern Recognition*, H. Lu, M. Blumenstein, S.-B. Cho, C.-L. Liu, Y. Yagi, and T. Kamiya, Eds. Cham: Springer Nature Switzerland, 2023, pp. 336–345.

[15] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[16] J. Švec and A. Frémund, "t5s - T5 made simple," 2022. [Online]. Available: https://github.com/honzas83/t5s

[17] A. Elkahky, K. Webster, D. Andor, and E. Pitler, "A challenge set and methods for noun-verb ambiguity," in *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 2562–2572.

[18] B. Bohnet, R. T. McDonald, G. Simões, D. Andor, E. Pitler, and J. Maynez, "Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings," in *Annual Meeting of the Association for Computational Linguistics*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:46896572

[19] M. Nicolis and V. Klimkov, "Homograph disambiguation with contextual word embeddings for TTS systems," in *Proc. ISCA Speech Synthesis Workshop (SSW 11)*, 2021, pp. 222–226.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. the 2019 North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, 2019.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. International Conference on Neural Information Processing Systems (NIPS)*, Long Beach, California, USA, 2017, p. 6000–6010.